

1. (original) A method of regulating TCP/IP connection requests which await service in a system by a TCP/IP connection control table to prevent overload thereof, said method comprising the steps of:

- a) monitoring usage of said system on a dynamic basis,
- b) based upon said usage, dynamically computing a time-out value T_{ho} which defines the time duration that a TCP connection request may await service by said system, and
- c) removing from said TCP/IP connection control table all TCP/IP connection requests which have been awaiting service in said TCP/IP stack for a duration exceeding T_{ho} .

2. (original) A method as set forth in Claim 1, wherein said TCP/IP connection control table has size N_{size} and an upper bound for usable table size of $N_{abs} \leq N_{size}$, and where values of T_{ho} are dynamically computed in a range $[T_{min}, T_{max}]$.

3. (original) A method as set forth in Claim 2, comprising the steps of:

- i) setting $T_{ho} = T_{min}$ when $N > N_{abs}$,
- ii) when $N > N_{limit}$ setting $T_{ho} = \max \{T_{min}, T'_{ho}/A\}$, where T'_{ho} is a previously existing value of T_{ho} , where $A > 1$, where N is the current usage of the table, and where $0 \leq N_{limit} \leq N_{size}$, and
- iii) when $N \leq N_{limit}$, setting $T_{ho} = \min\{T_{max}, A * T'_{ho}\}$.

4. (original) A method as set forth in Claim 2, comprising the steps of:

- a) defining a plurality of table usage value N_i spanning an increasing range of $N_i = 0$ to $N_i = N_{size}$,
- b) associating a corresponding plurality of time durations T_i spanning a decreasing range of $T_i = T_{max}$ to $T_i = T_{min}$, and
- c) comparing current table usage N to N_i and setting T_{ho} to a corresponding value T_i .

5. (currently amended) A method as set forth in Claim 2, wherein T_{\min} has a value in a range of 0.01 to ~~T~~ 1.0 secs. and wherein T_{\max} has a value in a range of 60 to 120 secs.

6. (new) A method of regulating TCP/IP connection requests which await service in a system by a TCP/IP connection control table to prevent overload thereof, said method comprising the steps of:

monitoring usage of said system on a dynamic basis,

based upon said usage, dynamically computing a time-out value T_{ho} which defines the time duration that a TCP connection request may await service by said system,

removing from said TCP/IP connection control table all TCP/IP connection requests which have been awaiting service in said TCP/IP stack for a duration exceeding T_{ho} ; and

setting $T_{ho} = T_{\min}$ when $N > N_{abs}$, when $N > N_{limit}$ setting $T_{ho} = \max \{T_{\min}, T'_{ho}/A\}$, where T'_{ho} is a previously existing value of T_{ho} , where $A > 1$, where N is the current usage of the table, and where $0 \leq N_{limit} \leq N_{size}$, and when $N \leq N_{limit}$, setting $T_{ho} = \min\{T_{\max}, A * T'_{ho}\}$;

wherein said TCP/IP connection control table has size N_{size} and an upper bound for usable table size of $N_{abs} \leq N_{size}$, and where values of T_{ho} are dynamically computed in a range $[T_{\min}, T_{\max}]$.

7. (new) A method as set forth in Claim 6, comprising the steps of:

- a) defining a plurality of table usage value N_i spanning an increasing range of $N_i = 0$ to $N_i = N_{size}$,
- b) associating a corresponding plurality of time durations T_i spanning a decreasing range of $T_i = T_{\max}$ to $T_i = T_{\min}$, and
- c) comparing current table usage N to N_i and setting T_{ho} to a corresponding value T_i .